REDUCE COSTS WITH MULTIMISSION SEQUENCING AND A MULTIMISSION OPERATIONS SYSTEM

David A. Bliss
Luis C. Morales
Jet Propulsion Laboratory/California Institute of Technology
Pasadena, California

ABSTRACT

Mission sequencing involves merging science and engineering inputs into an integrated, constraint-checked sequence and producing review and spacecraft command products. This task employs processes, procedures, and tools which have a high degree of commonality across all missions. The JPL Multi-Mission Office (MMO) Mission Planning and Sequencing Team (MPST) has successfully baselined these processes, procedures, and tools so that they are readily adaptable to missions of varying complexity. As a result, the MPST can quickly assemble a team that provides mission sequencing to very different missions at a fraction of previous costs. This paper will discuss the MMO MPST approach of adapting core processes, procedures, and tools to multiple missions. The paper will then propose extending this multi-mission philosophy to skeleton timeline development, science sequencing, and spacecraft sequencing. Finally, the paper will investigate a multi-mission approach to MOS development.

Multi-Mission Sequencing Operations

In general, missions have a process by which science observations are built and commanded. Missions have another process by which the spacecraft team builds and commands engineering events. And a sequencing team takes inputs from science and engineering event generation and merges them into an integrated product with constraint checks. The sequence team also translates the merged product into spacecraft commands. A high level description of and tools used for the four sequence processes of skeleton timeline development, science sequencing, spacecraft sequencing, and sequence integration steps are summarized below. These steps and tools are characteristic of all missions of varying complexity.

I. Skeleton timeline development:

Description: Skeletal timeline development produces the backbone upon which subsequent science observations and engineering events are placed. Tools such as APGEN insert DSN passes and critical engineering and navigation events into a timeline according to a skeleton timeline development operations process. For standard missions, skeleton timeline development and the following science sequence generation steps can be much reduced and sometimes eliminated, with science and engineering teams delivering straight to the sequence team for sequence integration. Remaining operations processes and multi-mission tools can be employed with relatively small adaptations.

General process steps: Sequence boundary identification, identification of key navigation and engineering event windows, high level identification of science campaign or observation boundaries

Tools to accomplish those steps: SOA or Science Opportunity Analyzer (enables opportunity identification and preliminary design for science observations), APGEN (activity planning tool which can automatically schedule activities based on a set of rules), mission-specific scheduling program

II. Science sequencing:

Description: Science observation generation then uses science observation generation and constraint checking software to produce and integrate science observations into the skeleton timeline.

General process steps: Observation opportunity identification, observation command implementation, science observation integration, science instrument and observation-internal flight rule checks. Science observation integration.

Tools to accomplish those steps: SEQGEN (enables generation, modification, expansion, modeling, constraint checking of spacecraft commands), PDT or POINTER (for remote sensing observations requiring spacecraft pointing and/or target motion compensation), in-house tools that meet SEQGEN SIS requirements.

III. Spacecraft/Engineering sequencing:

Description: Spacecraft/Engineering sequencing uses engineering blocks and commanded engineering events to operate, navigate, monitor, and maintain spacecraft systems and subsystems.

General process steps: Spacecraft system and subsystem operation and checkout. Optical Navigation and other Navigation events. Memory readout. Engineering event integration.

Tools to accomplish those steps: SEQGEN and other in-house tools that meet SEQGEN SIS requirements

IV. Sequence integration:

Description: Sequence integration and commanding uses multi-mission integration software to merge science observations and engineering events into a complete sequence, translate the sequence into spacecraft-readable command packets, perform constraint checks and memory management, and produce sequence review products. Real-time commanding uses multi-mission command generation software such as the Automated Sequence Processor (ASP) to generate real-time commands and real-time command mini-sequences for transmission to the spacecraft according to a multi-mission real-time command process. At present, only the this step has been made completely multi-mission by MPST.

General process steps: Merge Science and Spacecraft inputs into integrated sequence. Model merged sequence. Perform integration check, flight rule and constraint checks. Generate command uplink products

Tools to accomplish those steps: SEQGEN, SEQ_REVIEW (assists sequence product review, real-time and DSN support product generation), SEQTRAN (on Surveyor Bus based spacecraft, translates the SSF into S/C readable files and tracks memory usage and insures correct memory management), SLINC (converts SSF or UNIX binary file to Command Packet File format), CMD_TCWRAP (converts CPF into Spacecraft Modeling File), AUTOGEN (for highly repetitive sequencing, SASF inputs), Automated Command Tracker (tracks command and sequence status, review comments, provides required action notifications), Electronic Command Request Form (form ACT uses to record, review, and approve results).

To achieve extremely efficient multi-mission sequence operations, the MPST core mission programs, such as the ones identified above, enable them to handle mission specific adaptations via mission specific "adaptation" files that define the mission-specific commands, models, and constraint checks.

As an example, let us consider SEQGEN. SEQGEN allows a user to generate and modify requests, expand a series of requests into their resultant S/C commands, model these S/C commands, flag conflicts in the modeling of commands, flag violations of flight/mission rules, show the time extent of each request graphically, and graphically display model attributes. As implied above, SEQGEN consists of a multi-mission core program and a mission specific adaptation. The mission specific adaptation employs the following set of "adaptation" files that define the mission specific commands, models, and constraint checks:

a) Spacecraft Model File (SMF). The SMF contains the definition of spacecraft and ground subsystem models, and spacecraft command/parameter definitions
b) Flight/Mission Rules File (FMRF). The FMRF contains flight and mission rule checking algorithms
c) Spacecraft Activity Type File (SATF). Contains names and definitions of the activity types, including on-board blocks, ground expanded blocks, SEQGEN directives and SLINC directives
d) Context Variable (Definition) File (CVF). Contains parameters defined during the adaptation process that are used in the definition of activity types or models
e) Legend File. Contains data to define display definitions and layout

SEQGEN Inputs

SEQGEN requires the following input files to perform sequence expansion and constraint checking:

a) Spacecraft Clock Coefficient File (SCLK)
b) Orbit Propagation Timing and Geometry File (OPTG)
c) Lighttime File (LTF)
d) DSN Viewperiod File (VP)
e) Viewperiod Format Description File (VIEW_FD)
f) DSN Station Allocation File (SAF)
g) Initial Conditions File (INCON)
h) Context Variable File (CVF)
i) Spacecraft Activity Sequence File (SASF)
j) Spacecraft Activity Type File (SATF)

SEQGEN Outputs

a) Spacecraft Activity Sequence File (SASF)
b) Spacecraft Sequence File (SSF)
c) Predicted Events File (PEF)
d) Final Conditions File (FINCON)
e) Run Log

Figures 1 (Ref.1) shows SEQGEN inputs and outputs. In both cases, note the input of the SMF, FMRM, and SATF files "adaptation" files. Figure 2 (Ref. 2) shows the overall flowchart for the set of core Mission Services and Applications (MS&A) software which is adaptable to support multiple missions.

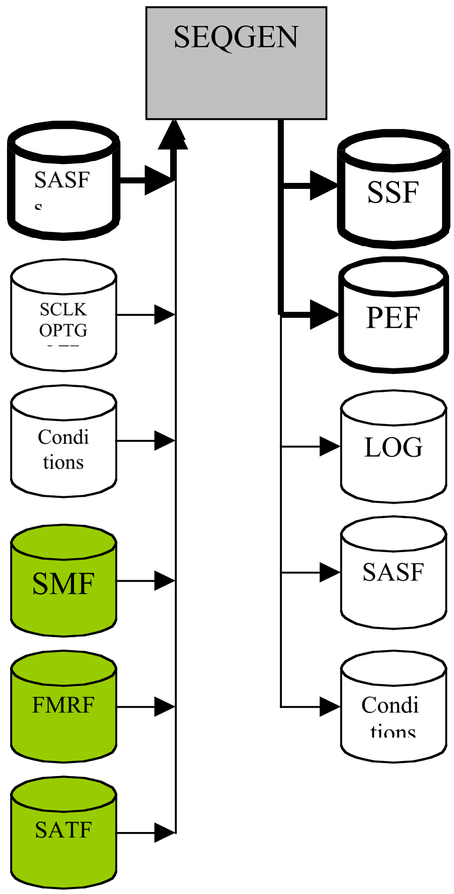

Figure 1: Uplink Data Flow for Surveyor Bus

In addition, the MPST has "wrapped" these adaptable, core tools so that their use enables a consistent, multi-mission process. These "wrappers" are scripts which use tables that define states for different spacecraft. For example, when performing sequence integration, the MPST user identifies the spacecraft number. The script will then reference the spacecraft data tables corresponding to the spacecraft number. These data tables then tell the script what to do for the identified spacecraft. So if a user is running SLINC, the script will use a specified spacecraft number to consult a table and determine whether the spacecraft number refers to a VML spacecraft. If so, the script runs a VML compiler. If not, it executes another routine which involves memory management.

By cross-training personnel to use these wrapped, adaptable, core tools, six MPST engineers are successfully able to provide full sequence integration and command generation services to four launched missions of mostly standard to medium complexity (MGS, Genesis, Odyssey, and Stardust). These numbers are much lower than what was required in the past.

A similar approach should be employable for skeleton timeline development, science sequencing, and spacecraft/engineering sequencing. The tools SOA, APGEN, SEQGEN (as described above), and POINTER all represent core programs which could be adaptable to different missions. During the latter phases of science and spacecraft/engineering sequencing, when commands are developed, scripts could wrap the core programs to execute within the context of the pertinent mission spacecraft. Such as approach could enable the development of multi-mission science and spacecraft sequencing capability.

Overall, sequence operations costs are influenced by mission complexity. Mission complexity is driven by spacecraft pointing requirements, navigation requirements, unique observing requirements, spacecraft landing requirements (if any), payload data acquisition capability, environmental constraints, spacecraft downlink capability, and new spacecraft technology. These factors determine whether complexity can be classified as standard, medium, or complex.

1) Standard missions do not have strict and tight pointing requirements. They do not employ new technology on critical subsystems. Operations are repetitive (such as a mapping mission). Standard missions have no lander or target contact aspects to the mission. For standard missions, teams (such as science integration and sequencing), can be combined and the number of required MOS components can be reduced. Remaining MOS components can be employed with relatively small adaptations. Initial costing estimates would be performed based upon the reduced set of MOS components used and then adjusted according to the number of small adaptations.

2) Missions of medium complexity can have precise pointing requirements but not motion compensation. They can have new technology on one critical subsystem which is well-tested and well-modeled. Involved missions can have non-repetitive operations. Involved missions can involve distant target contact such as firing at a target to analyze ejecta or touching a target surface with a sample arm. For involved missions, teams (such as science integration and sequencing), can be combined and the number of required MOS components can be reduced. Required MOS components can be employed with moderate adaptations. For involved operations, one or both sequence development steps 1 and 2 can be eliminated with the other step(s) existing on a much reduced level. Remaining operations processes and multi-mission tools can be employed with moderate adaptations. Initial costing estimates would be performed based upon the number of MOS components used and then adjusted according to the number of moderate adaptations.

3) Complex missions have precise pointing requirements with motion compensation. They can have new technology

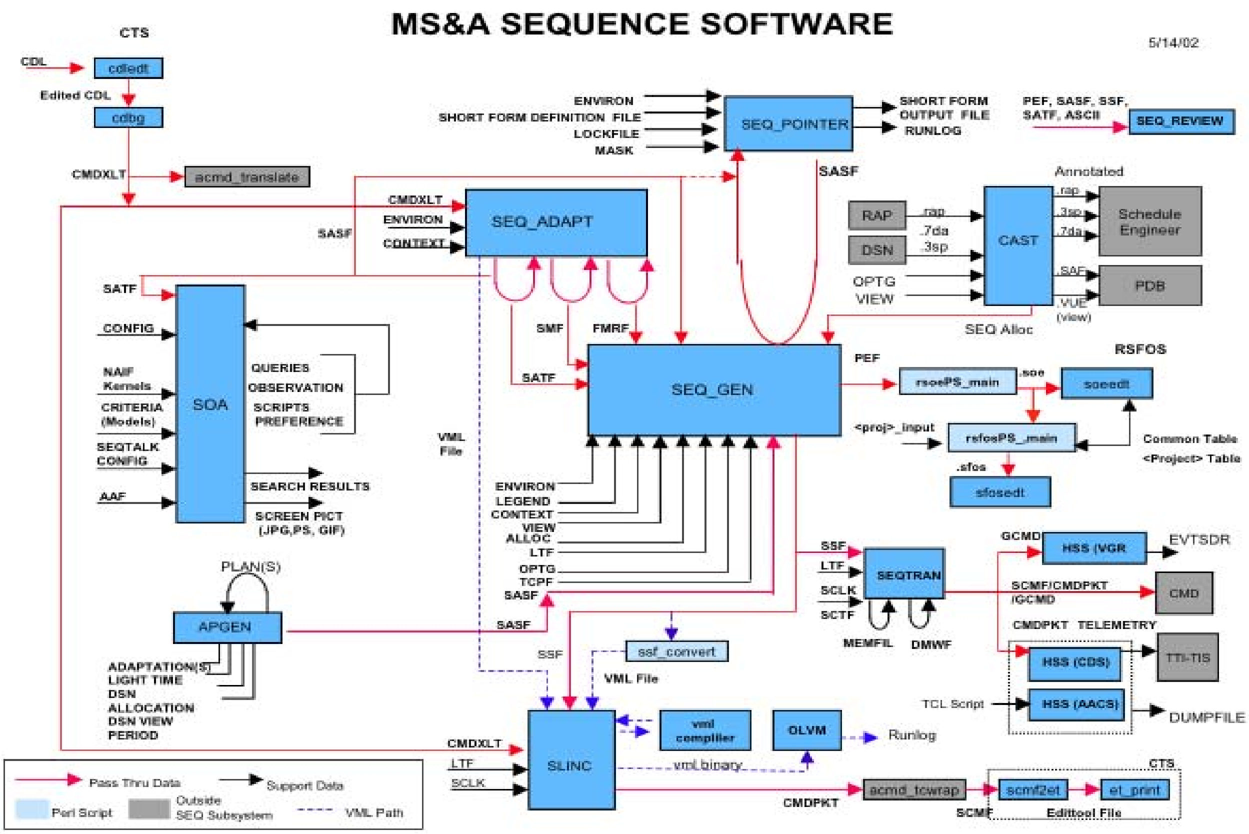

Figure 2: MS&A Software

on one or more critical subsystems. They can have multiple, unique observation designs. The main spacecraft body may be a lander. Complex missions most likely will require all MOS components. MOS components can be employed with extensive adaptations. For complex missions, all of the above sequence development steps are likely to be required. Operations processes and multi-mission tools can be employed with extensive adaptations and tests. Initial costing estimates would be performed based upon the complete set of MOS components used and then adjusted according to the number of extensive adaptations.

The more complex a mission is, the more complex the tool adaptations (ie, pointing models, navigation requirements, subsystem modeling requirements) tend to be. In addition, mission complexity determines how may of the previously mentioned sequence processes a project must use.

Complex missions often require all four of the above sequence development processes. Missions of medium complexity may enable projects to combine the skeleton timeline development and science sequencing processes. Missions of standard complexity may enable teams, such as Genesis and Stardust, to bypass the skeleton timeline and science sequencing processes completely and have science teams deliver directly to the sequence team for sequence integration. Further, for repetitive mapping missions such as Odyssey and Mars Global Surveyor, the mission sequence process can be simplified. Tools such as AUTOGEN can retrieve input files from data repositories and automatically generate spacecraft sequences based upon a set of rules.

Multi-Mission MOS Baseline

In general, an MOS must provide the personnel, procedures, facilities, hardware and software required to conduct mission operations. The following thirteen uplink and downlink functional areas are characteristic of an MOS:

Uplink	Downlink
Misson Planning	Mission Monitor & Control
Science Planning	Tracking Data Analysis
Science Sequencing	Telemetry Data Processing
Mission Sequencing	Navigation
Command Processing	Data Management & Archive
Simulation	Flight System Analysis
	Science Data Products

The Deep Space Network (DSN) and the Deep Space Mission System (DSMS) services for handling telemetry, command, and radio metric data are already multi-mission. This paper proposes establishing a multi-mission MOS baseline by making other, Level 3 capabilities multi-mission. The following capabilities are proposed for multi-mission construction.

Standardized General Requirements: Ground Segment Verification, Validation, Training, and Configuration Management

General Uplink Requirements: Short Term Scheduling and Sequence Packaging; Ancillary File Generation and Data Production; Mission Planning; Science Planning; Sequence Generation, Validation, Approval, and Update; Adaptability; Block Generation; Pointing Operations; Real Time Commanding; Anomaly Response; Maneuver Generation; Command Radiation; Packet Acknowledgement Process; DSN Scheduling; Block Development Process; Restricted Command Process

General Downlink Requirements: DSN Data Capture; Telemetry Processing; Real Time Monitoring; Non Real-Time Analysis; Frame Reconstruction; Instrument/Payload Performance Analysis; Science Data Processing and Analysis; Tracking and Navigation; Data Collection and Processing; Data Archive

General Flight Rules and Team Checking Responsibility:

Flight Rule	Responsible Checking Team(s)
Command and Data Handling;	Spacecraft, Sequencing
Fault Protection	Spacecraft
Electronics	Spacecraft
Flight Software	Spacecraft
Instrumentation	Science, Sequencing
Attitude and Articulation/Pointing	Spacecraft, Sequencing
Power Generation and Distribution	Spacecraft
Reaction Control System	Spacecraft
Structure and Mechanical Subsystem	Spacecraft
Thermal Control Subsystem	Spacecraft
Telecommunications	Spacecraft
Spacecraft	Spacecraft
Virtual Machine	Spacecraft, Sequencing

Operational Interface Agreements (OIAs)

Instrument Operations, Real-Time Operations, Uplink Operations, Navigation, Spacecraft Operations, System Engineering, Radio Science, Navigation, DSN Services, Mission Sequencing, Science Sequencing, Mission Planning, Simulation and Verification, Programmatic Management

Software Interface Specifications (SISs)

Instrument Operations, Real-Time Operations, Uplink Operations, Navigation, Spacecraft Operations, System Engineering, Radio Science, Navigation, DSN Services, Mission Sequencing, Science Sequencing, Mission Planning, Simulation and Verification, Programmatic Management

Processes

Science and Mission Planning processes, sequencing processes, simulation process, navigation process, flight system analysis process, science data products process, and archive process.

The aforementioned requirements, flight rules, OIAs, and SISs can be maintained as an MOS Baseline within a multi-mission database. The database maintains team responsibilities for each MOS component (ie, each flight team is assigned a set of requirements, OIAs, SISs, flight rules, and procedures that it is responsible for adapting and following). The multi-mission database could be used to update the MOS baseline. Information contained within the MOS baseline would serve as a basis for costing, implementation, and scheduling.

For costing, each set of MOS baseline components would include costs based upon recent project experience (ie, recent project costs to satisfy baseline real-time command costs, fault protection flight rules, Navigation OIAs, Mission Sequencing SISs, etc). As part of their early development, projects could extract the MOS baseline from the multi-mission database and apply the baseline costs as a starting point. Projects could then add details and modifications to requirements, flight rules, OIAs, SISs, and processes. These modifications would then be used to adjust costs.

Within the JPL Multi-Mission Office, a multi-mission GDS already exists. In its early development phases and based upon MOS and other needs, a project submits an adaptation to the multi-mission core baseline. A ground data system engineer (GDSE) coordinates implementation of the adaptation with the multi-mission GDS baseline. Once the adaptation is implemented and tested, it is placed upon a server for the client project to download. From one server, four projects (Odyssey, Genesis, MGS, and Stardust) download their own adaptation of the multi-mission core GDS. The proposed, multi-mission MOS could be constructed so that it could work, with adaptation, from the multi-mission GDS. With such a design, the multi-mission GDS would contain tools and software to support the multi-mission MOS functions.

For MOS adaptation scheduling, an automated multi-mission scheduling program (perhaps EXCEL) could lay out a schedule for the MOS component development based upon dates for Launch and key mission reviews. Launch and key mission reviews, such as the launch readiness review, critical design review, preliminary design review, etc) drive the dates when adapted MOS components (ie. OIAs, SISs, Operational Readiness Tests (ORTs), mission plan, team plans and procedures, etc) are due. An algorithm within the scheduling program would then determine when final adaptations must be begun and completed. Obviously, for complex missions, more time would have to be allowed for the adaptation and finalization of MOS components than for missions of medium complexity. The program would also allow some development margin (10% to 15%). The program could then plot an entire MOS adaptation and finalization development timeline. The automated multi-mission scheduling program would greatly reduce the amount of manual scheduling work required of an MOS Engineer.

Upon start of a mission, an MOS Engineer could withdraw from the MOS Baseline database the MOS components characteristic of the project's complexity and obtain a baseline cost. From one or a few project interactions, the MOS Engineer could develop a first cut of the MOS components adaptation and cost. Using project launch and key review dates, the MOS Engineer could generate the final adaptation schedule. Included in this schedule would be the date when the adapted multi-mission GDS would be available with the software and tools to support MOS functions

Historically, planning, costing, implementing, and validating a project mission operations system has sometimes required years. An MOS Engineer trained in the use of the multi-mission MOS Baseline database and scheduling tool could cut this time considerably. For a standard mission requiring little adaptation, the MOS development timeline could be cut to weeks.

Acknowledgements:

A special thanks goes to Bruce Waggoner for his review of, critique of, and inputs to this paper. Thanks also to Vic Voskanian, Tim Weise, and Jenny Cruz for very helpful contributions and discussions.

References:

Ref. 1: "Uplink Process Tools, Inputs, & Products". A presentation by Reid Thomas and Bruce Waggoner

Ref. 2: "DSMS Mission Services and Applications SEQUENCE" website